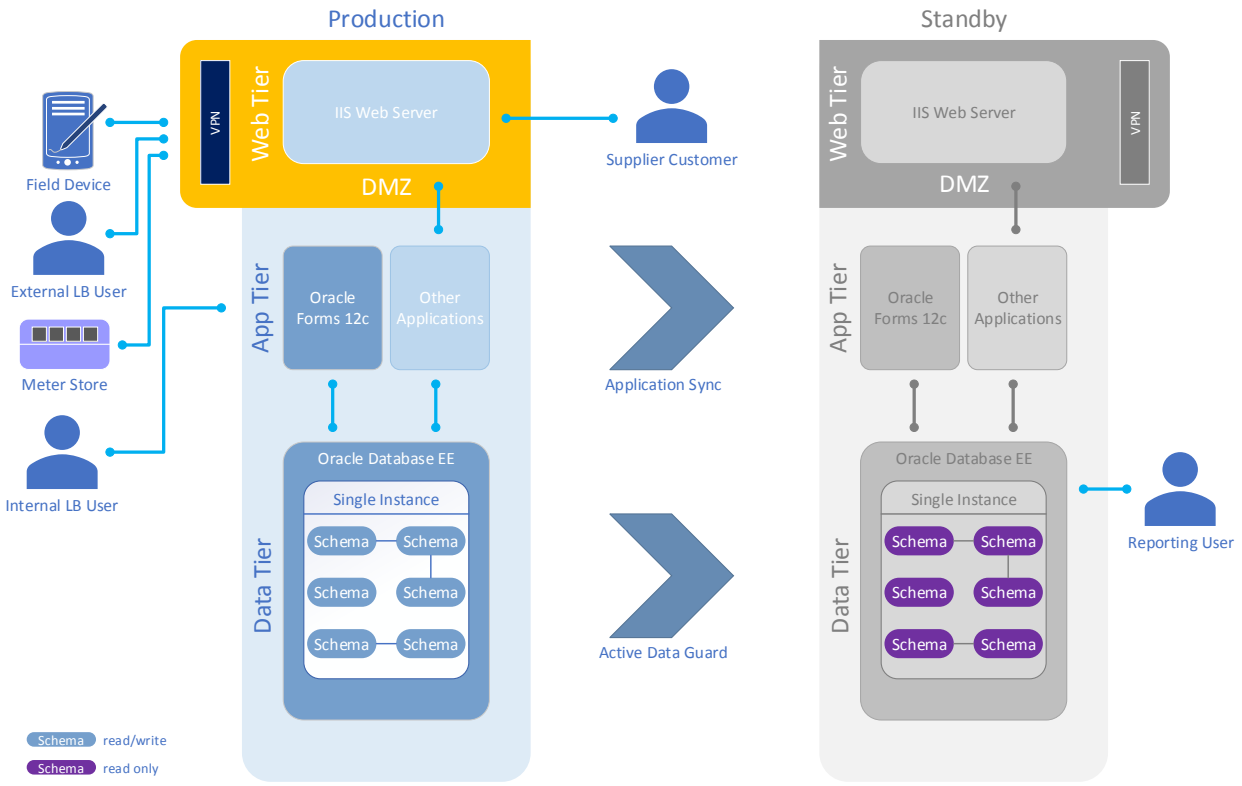
TECHNICAL ARCHITECTURE ASSESSMENT

Author: Zabair.Zafar@[client].co.uk

11th October 2019

OVERVIEW OF THE [CLIENT] CORE ARCHITECTURE

The diagram below provides a high-level view of the [CLIENT] core architecture stack:



Web Tier

- Virtualised single instance Microsoft IIS web server sitting within a DMZ (DeMilitarised Zone).
- The IIS web server services incoming connections from supplier customer, who use the [CLIENT] website to enter meter readings.
- External [CLIENT] users, field devices and meter stores connect via a VPN (Virtual Private Network) connection.

Application Tier

- Virtualised single instance Oracle Forms 12c application tier.
- Active-Passive, with standby site.
- DNS re-direct to standby in the event of Production failure (only for some of the applications)

Database Tier

- Virtualised single instance Oracle Database tier. Oracle RAC has been considered in the past (cost outweighed the benefits).
- Active-Passive, with standby site.
- Oracle Database Enterprise Edition.

- The database is about 2TB in size.
- One Oracle database instance contains many application schemas.
- Processor-based licensing (purchased through a UK vendor, Explorer(link)).
- Peak performance is currently at 50-60% utilisation.

Standby Site

- Standby data replication using Oracle Active Data Guard.
- Standby database used for reporting.
- Scaled to 50% of Production.
- Database auto failover to standby site in the event of Production failure.

OPPORTUNITIES FOR IMPROVEMENTS & QUICK WINS

[CLIENT] DATA MODEL

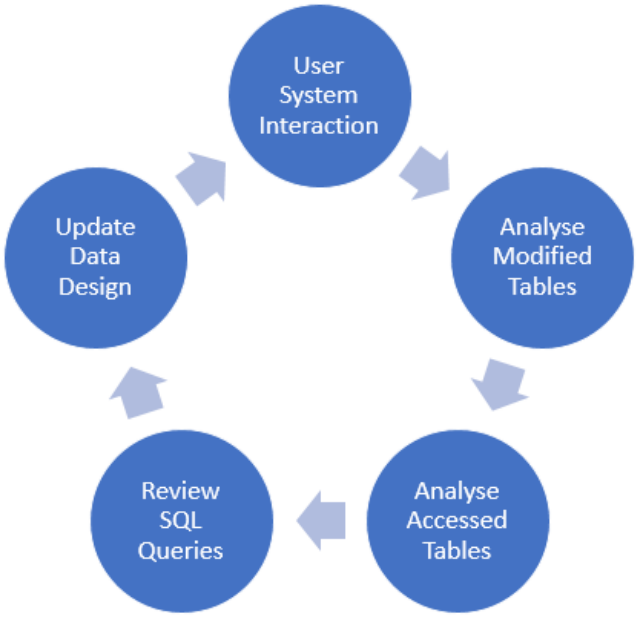
The core [CLIENT] data model and schemas have organically grown over the past 20 years, with very little governance or control over the structure and usage. Tables have been created and queries have been written in isolation. Referential integrity has not been maintained, which has led to duplication of data and broken relationships between tables. [CLIENT] is currently faced with the following challenges:

1. Duplicate data across schemas.
2. Uncertainty around table relationships due to broken referential integrity.
3. Uncertainty around which tables are currently being used and which are obsolete.
4. Due to points 2 and 3, [CLIENT] is unable to delete or archive data.

It's absolutely critical that [CLIENT] understands its data before any transformation programme can begin. Otherwise data migration, data archiving, data retention and GDPR compliance will pose a significant challenge.

Recommendations

There are no Oracle tools available that will allow [CLIENT] to identify data structures and their current usage. There are tools that allow the reverse engineering of database schemas. However, these are not sufficient for the task at hand. As illustrated in the diagram below, in order to understand the [CLIENT] As-Is data, an incremental and cyclical approach is required:



1. **User System Interaction**: While users are interacting with the core system, it is possible to capture SQL queries that are being executed against the Oracle database. Knowing this information can allow us to slowly and incrementally *'paint a picture'* of the data structures being used (and not used).
2. **Analyse modified tables:** An Oracle database already captures the date and time of when a table was last modified using UPDATE, INSERT and DELETE SQL statements. This information is stored in Oracle database views.
3. **Analyse accessed tables:** Oracle doesn't store data for tables that were last accessed using the SELECT SQL statement. Auditing of tables needs to be enabled. This can be done progressively on selected tables to ensure database performance isn't impacted.
4. **Review SQL Queries:** Review core application queries / stored procedures.
5. **Update Data Design:** A data ER (Entity Relationship) design(s) can be updated once enough information has been gathered.

It's important to repeat the above cycle while users continue to interact with the system. This will ensure enough variation of user activity has occurred to comprehensively capture database usage.

Quick Win

[CLIENT] acknowledge that in order to understand the data, it's useful for an external independent entity to undertake this task.

The recommended approach outlined above can be completed by a full-time resource within 3-6 months. Considering the criticality of understanding the data, the successful completion of this task will add huge value to [CLIENT] and therefore should be considered a significant quick win.

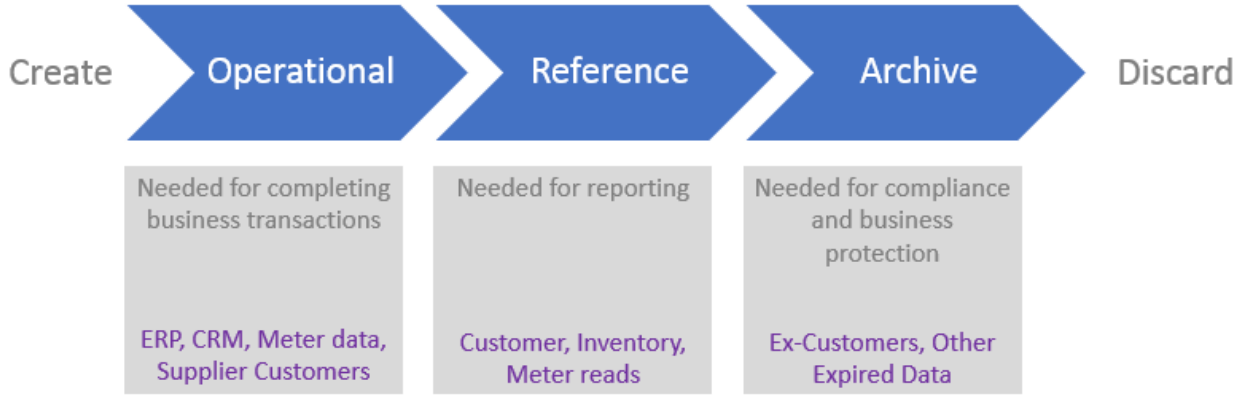
DATA RETENTION & ARCHIVING

[CLIENT] is required to archive personal data of customers once a contract with a utilities company ends. The removal of personal customer data has become more of an immediate concern due to GDPR implications. At present, there is no process or mechanism in place to remove personal data of ex-customers.

From a GDPR perspective, there is also a need to remove/anonymise personal data of ex-employees and to provide details of personal data upon request. This is difficult to do considering the state of the [CLIENT] data.

Recommendations

There needs to be greater clarity around how data is used as it matures and progresses through the data lifecycle:



In order for [CLIENT] to start thinking about the data life cycle outlined above the following needs to be done:

1. Understand the As-Is data: As stated previously, it's vital that [CLIENT] understands its data before initiating any data retention and archiving activities. [CLIENT] can not delete or archive its data unless it first understands the As-Is data sources.
2. Enforce referential integrity once the As-Is data state in known. Enforcing referential integrity rules ensure that related tables remain in a consistent state relative to each other.
3. Formulate a data retention policy for the different categories of data held within [CLIENT]. A data retention policy is an organisation's established protocol for retaining information for operational or regulatory compliance needs. A comprehensive data retention policy outlines the business reasons for retaining specific data as well as what to do with it when targeted for disposal. I believe [CLIENT] is currently in the process of producing such a document.

One Oracle technology that can help [CLIENT] with archiving old data, not necessarily ex-customer data, is Oracle partitioning:

Oracle Partitioning: Partitioning allows tables, indexes, and index-organized tables to be subdivided into smaller pieces, enabling these database objects to be managed and accessed at a finer level of granularity. Oracle Partitioning is a separately licensed option of the Oracle Database Enterprise Edition. The list price is about £9,000 per processor core.

DATA PROTECTION

[CLIENT] are concerned about the protection of the personal data held within their database. Unless corrective measures are taken, there is a risk that personal data of customers and employees could be compromised.

At present, [CLIENT] does not encrypt internal network traffic or data at rest. There is also a need to review their role-based access to ensure the right people have the right level of access. Operational support personnel potentially have access to personal data, which isn't necessarily required to fulfil their job.

Recommendations

There are a number of Oracle tools available that can assist [CLIENT] in protecting the personal data residing in their database. I believe some of these tools have already been considered by [CLIENT] in the past, but it is unclear why these particular technologies were not implemented:

1. **Encrypting data at rest using TDE (Transparent Data Encryption):** TDE helps protect data stored on media (also called data at rest) in the event that the storage media or data file is stolen.
2. **Encrypting database backups using RMAN (Recovery Manager):** For improved security, RMAN backups created as backup sets can be encrypted. Encrypted backups cannot be read if they are obtained by unauthorized people.
3. **Controlling data access using Database Vault:** Oracle Database Vault provides powerful cyber security controls to help protect application data from unauthorized access and improve compliance with privacy and regulatory requirements.

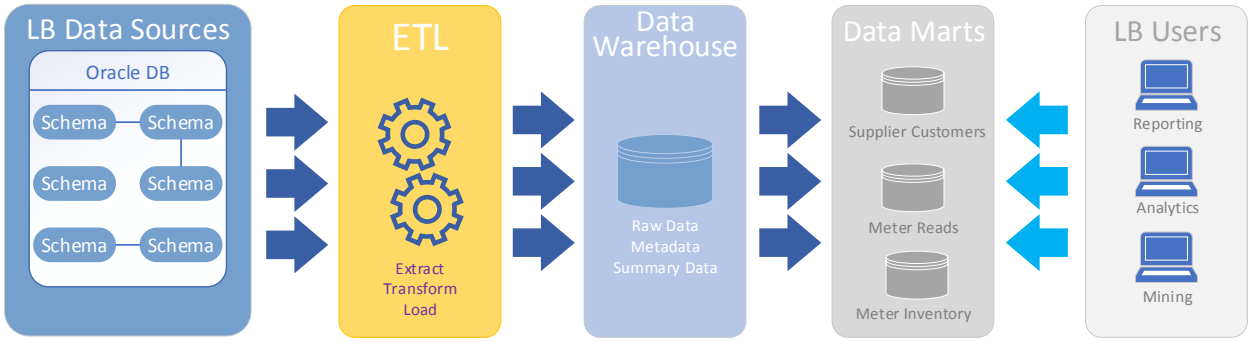
Quick Wins

- Implement TDE to encrypt data at rest.
- Review existing Role-Based access to the database and implement tighter controllers.

The quick wins outlined above can be completed by a full-time resource within 3-6 months. TDE is a licensed feature, which comes with the Advanced Security pack. The list price is about £12,000 per processor core.

BI REPORTING

[CLIENT] currently uses the standby database for reporting. This is made possible through the use of Active Data Guard, which allows read-only real-time query executions. Using the standby database for reporting purposes is standard practice as it reduces the resource demands placed on the Production database. However, the reporting tables have been created as and when a reporting requirement comes along. This tactical approach in creating reporting tables means there is now a lot of unstructured data tables with potentially considerable duplication of data. There is now a need to move towards a more strategic view by adopting a Data Warehouse/Mart/Lake model for BI reporting:



[CLIENT] Data Sources are the many schemas sitting in the [CLIENT] Central database.

ETL (Extract Transform Load) layer copies data from the source, transforms the data (if necessary) and moves it to the destination i.e. the Data Warehouse. Unless [CLIENT] can identify its source data, it's not going to be possible to define an ETL layer.

Data Warehouse is a large centralised repository of data that contains information from many sources within an organisation. The collated data can be used to populate specialist Data Marts.

Data Marts contain repositories of summarised data collected for analysis on a specific section or unit within an organization, for example, meter inventory.

Recommendations

- Understand the As-Is data: As stated previously, it's vital that [CLIENT] understands its As-Is data before undertaking the To-Be activities around Data Warehousing.
- Begin working on a BI Data Warehouse/Lake model design.

Quick Wins

- Begin working on a BI Data Warehouse/Lake model design.
- The quick win can be completed by a full-time resource within 3-6 months.
- There is dependency on successfully defining the [CLIENT] As-Is data.